



Algorithms and Data Structures – Conditional Course Winter Term 2026/27

Exercise Sheet 9

Due: Wednesday, 23 December 2026, 12:00

Exercise 1: Topological Sorting

(6 Points)

- (a) Draw the directed graph G defined by this adjacency matrix.

(2 Points)

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

- (b) Give a topological ordering of G .

(1 Point)

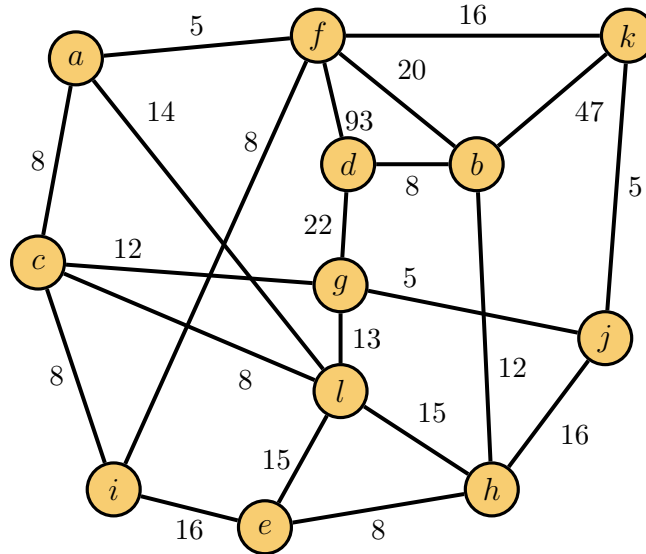
- (c) How many topological orderings does G have?

(2 Points)

- (d) Let $G = (V, E)$ and $G' = (V, E')$ be directed acyclic graphs on the same vertex set, with $E' \subseteq E$. How does the number of topological orderings of G compare with that of G' ? Justify your answer. Use a statement such as “equal”, “strictly greater”, “at least as many”, or “at most as many”.
(1 Point)

Exercise 2: Prim's Algorithm

(4 Points)



Run Prim's algorithm to compute a minimum spanning tree (MST).

- Start at vertex a .
- Break ties between eligible edges (u, v) , where $u \in A$ and $v \notin A$, lexicographically. For example, choose (a, c) before (f, i) , and (c, i) before (f, i) .
- Mark the edges of the final spanning tree.
- List the order in which the edges are added.
- Give the total weight of the final MST.

Exercise 3: Unique Minimum Spanning Trees

(10 Points)

Let $G = (V, E, w)$ be an undirected, connected, weighted graph.

(a) Prove that if all edge weights are pairwise distinct, G has a unique minimum spanning tree. (5 Points)

(b) Prove that the following construction produces an MST T' of G : (5 Points)

Start with $T' = (V, E)$. Find a cycle C in T' and delete a maximum-weight edge of C . Repeat until T' contains no cycles.

Exercise 4: Bonus: The Travelling Salesperson Problem (7* Points)

Let $p_1, \dots, p_n \in \mathbb{R}^2$ be points representing cities in the Euclidean plane. The distance between cities i and j is the Euclidean distance between p_i and p_j . A tour visits each city once and returns to its starting city. We seek a tour of minimum total length. This problem is hard in general,¹ so we settle for a tour at most twice as long as an optimal one.

Model this as a complete undirected graph $G = (V, E, w)$ with $V = \{p_1, \dots, p_n\}$ and $w(p_i, p_j) = \|p_i - p_j\|_2$. Its weights satisfy the triangle inequality:² the length of a tour $(i_1, \dots, i_n)$ is

$$w(p_{i_n}, p_{i_1}) + \sum_{j=1}^{n-1} w(p_{i_j}, p_{i_{j+1}}).$$

We will show that visiting the vertices of an MST in preorder, starting at any root, gives a tour no more than twice as long as an optimal tour.³

Let $R = (i_1, \dots, i_n)$ be an optimal tour of weight $w(R)$, and let T be an MST of weight $W_T = \sum_{e \in T} w(e)$. Let $P_T = (\ell_1, \dots, \ell_n)$ be the preorder traversal of T . Since G is complete, this also specifies a tour: follow the direct edges $\ell_1 \rightarrow \ell_2 \rightarrow \dots \rightarrow \ell_n \rightarrow \ell_1$, rather than walking along tree edges. Its weight is

$$W_{PRE} = w(p_{\ell_n}, p_{\ell_1}) + \sum_{j=1}^{n-1} w(p_{\ell_j}, p_{\ell_{j+1}}).$$

(a) Show that $W_{PRE} \leq 2W_T$. (5 Bonus Points)

(b) Show that $W_T \leq w(R)$. (2 Bonus Points)

The claimed approximation follows from $W_{PRE} \leq 2W_T \leq 2w(R)$.

¹The decision version of the exact travelling salesperson problem is NP-complete. A polynomial-time algorithm is not known; one would imply $\mathcal{P} = \mathcal{NP}$.

²For all vertices a, b, c , $w(a, b) \leq w(a, c) + w(c, b)$: a direct edge is no longer than a detour.

³Preorder generalizes from binary trees: visit the root, then recursively visit its subtrees from left to right.